LISTING OF CLAIMS

Claims

1.   (Currently Amended) A computer-implemented method for optimizing transactional behavior of a middle-tier server between a client application and a database-tier server, the method comprising:

a middle-tier server remote from a client application creating a transaction policy on the middle-tier server by translating a deployment descriptor file;

a database-tier server remote from the client application creating a transaction policy on the database-tier server by translating the deployment descriptor file;

the client application calling a CORBA method, wherein the client resides on a system local to the client, wherein the CORBA method resides on a database-tier server remote from the client, and wherein the call comprises an Internet Inter-ORB Protocol ("IIOP") message sent on a path to the CORBA method on a database-tier server wherein the IIOP message includes a method name for the CORBA method called;

an interceptor residing on the middle-tier server intercepting the IIOP message;

the interceptor residing on the middle-tier server checking the transaction policy for the tier status of the server;

the interceptor residing on the middle-tier server returning the IIOP message to its path towards the CORBA method without completing a control object interpositioning process;

an interceptor residing on the database-tier server intercepting the IIOP message after it has passed through the middle-tier server;

2

the interceptor residing on the database-tier server checking the transaction policy for the tier status of the server;

the interceptor residing on the database-tier server checking the transaction policy for the database-tier server with respect to the method name;

the interceptor residing on the database-tier server either invoking the called CORBA method directly or first completing a control object interpositioning process between the object representing the transaction context and an Object Transaction Service spanning both the system local to the client and the database-tier server and then invoking the called CORBA method where the choice is defined by the results of the check of the transaction policy with respect to the method name.

2.   (Cancelled)

3.   (Original)   The method of claim 1, wherein the transaction policy created on the middle-tier server is created during deployment of the middle-tier server.

4.   (Original)   The method of claim 1, wherein the transaction policy created on the middle-tier server is created after receipt of the IIOP message to facilitate run-time checking of the deployment descriptor file to determine the transaction policy for the server.

3

18349.03/4000.01900

5.   (Currently Amended) A <u>computer-implemented</u> method for changing transactional behavior for a server; the method comprising:

defining transactional behavior for a server in a <u>first</u> transaction policy implemented on the server <u>by translating,</u> ~~wherein~~ the <u>first</u> transaction policy ~~is translated~~ from a deployment descriptor file during deployment of the server<u>;</u>

<u>client objects invoking,</u> ~~and wherein invocations of~~ a CORBA method ~~from client objects~~ resul<u>ting</u> in a <u>first</u> defined transactional behavior based on the <u>first</u> transaction policy;

modifying the deployment descriptor file <u>to a modified deployment descriptor file</u> to change the transactional behavior for the server;

redeploying the server <u>including</u> ~~which~~ implement~~s~~<u>ing</u> a modified transaction policy translated from the modified deployment descriptor file<u>;</u>

<u>identical client objects employing</u> ~~wherein~~ identical invocations ~~from identical client objects~~ resul<u>ting</u> in a <u>second</u> ~~different~~ defined transactional behavior for the server based on the modified transaction policy <u>which is different from the first defined transactional behavior for the server based on the first transaction policy;</u>

<u>wherein a negative transaction policy for the server results in one of the defined transactional behaviors which comprises a pass through of the CORBA method invoked without completing a control object interpositioning process; and,</u>

<u>wherein a positive transaction policy for the server results in the other of the defined transactional behaviors which comprises completing a control object interpositioning process for the CORBA method invoked.</u>

4

6.   (Cancelled)

7.   (Currently Amended) <u>A computer-implemented method for changing transactional behavior for a server; the method comprising:</u>

<u>defining transactional behavior for a server in a first transaction policy implemented on the server by translating the first transaction policy from a deployment descriptor file during deployment of the server;</u>

<u>client objects invoking a CORBA method resulting in a first defined transactional behavior based on the first transaction policy;</u>

<u>modifying the deployment descriptor file to a modified deployment descriptor file to change the transactional behavior for the server;</u>

<u>redeploying the server including implementing a modified transaction policy translated from the modified deployment descriptor file;</u>

<u>identical client objects employing identical invocations resulting in a second defined transactional behavior for the server based on the modified transaction policy which is different from the first defined transactional behavior for the server based on the first transaction policy;</u>

~~The method of claim 5,~~ wherein the deployment descriptor file and the transaction policy translated from the deployment descriptor file define transactional behavior for at least one CORBA method resident on the server in addition to transactional behavior for the server;

5

wherein a negative transaction policy for the server results in <u>one of the defined</u>

<u>transactional behaviors which comprises</u> a pass through of all invocations of CORBA methods

without completing a control object interpositioning process; and,

wherein a positive transaction policy for the server results in <u>the other of the defined</u>

<u>transactional behaviors which comprises</u> checking the transaction policy with respect to the

specific CORBA method invoked to determine if a control object interpositioning process should

be completed.

8.   (Original)   The method of claim 7, wherein the deployment descriptor file and the

transaction policy translated from the deployment descriptor file define transactional behavior

for all CORBA methods resident on the server in addition to transactional behavior for the

server.

9.   (Original)   The method of claim 5, wherein the deployment descriptor file is stored on

the server.

10.   (Original)   The method of claim 5, wherein the deployment descriptor file is stored in

a location remote from the server.

11.   (Original)   The method of claim 10 wherein the deployment descriptor file is

translated by a plurality of servers to create the transaction policies for the plurality of servers.

6

Atty. Dkt. IDF 1539   *Patent*
    4000-01900

12.   (Currently Amended) A <u>computer-implemented</u> method for setting transactional behavior

for a middle-tier server between a client application and a database-tier server, the method

comprising:

        a middle-tier server remote from a client creating a transaction policy by translating a

deployment descriptor file during deployment of the server;

        a database-tier server remote from the client creating a transaction policy by translating a

deployment descriptor file during deployment of the server;

        the client calling a CORBA method, wherein the client resides on a system local to the

client, wherein the CORBA method resides on the database-tier server remote from the client,

wherein the call comprises an <u>Internet Inter-ORB Protocol ("IIOP")</u> message having a service

context, wherein the IIOP message is sent on a path to the CORBA method on the database-tier

server, wherein the path comprises the middle-tier server, and wherein the IIOP message

includes a method name for the CORBA method called;

        an interceptor intercepting the IIOP message, wherein the interceptor resides on the

system local to the client;

        the interceptor residing on the system local to the client inserting an object representing

the transaction context on the service context of the IIOP message;

        the interceptor residing on the system local to the client returning the IIOP message to its

original path;

        an interceptor residing on the middle-tier server remote from the client intercepting the

IIOP message;

7

18349.03/4000.01900

the interceptor residing on the middle-tier server checking the transaction policy for the middle-tier server with respect to the server;

the interceptor residing on the middle-tier server returning the IIOP message to its original path without completing a control object interpositioning process;

an interceptor residing on the database-tier server remote from the client intercepting the IIOP message;

the interceptor residing on the database-tier server checking the transaction policy for the database-tier server with respect to the server;

the interceptor residing on the database-tier server extracting the object representing the transaction context from the service context of the IIOP message and reading the method name from the IIOP message;

the interceptor residing on the database-tier server checking the transaction policy for the database-tier server with respect to the method name;

the interceptor residing on the database-tier server either invoking the called CORBA method directly or first completing a control object interpositioning process between the object representing the transaction context and an OTS spanning both the system local to the client and the database-tier server and then invoking the called CORBA method where the choice is defined by the results of the check of the transaction policy with respect to the method name.

8

13.   (Original)   The method of claim 12, further comprising the interceptor residing on the middle-tier server extracting the object representing the transaction context from the service context of the IIOP message and reading the method name from the IIOP message after the interceptor intercepts the IIOP message and before the interceptor returns the IIOP message to its original path.

14.   (Original)   The method of claim 13, further comprising:

a second middle-tier server remote from the client on the path of the CORBA method invocation between the middle-tier server and the database-tier server;

the second middle-tier server creating a transaction policy by translating a deployment descriptor file during deployment of the server;

an interceptor residing on the second middle-tier server intercepting the IIOP message;

the interceptor residing on the second middle-tier server checking the transaction policy for the second middle-tier server with respect to the server;

the interceptor residing on the second middle-tier server extracting the object representing the transaction context from the service context of the IIOP message and reading the method name from the IIOP message;

the interceptor residing on the second middle-tier server returning the IIOP message to its original path without completing a control object interpositioning process.

9

15.   (Original)   The method of claim 13, further comprising:

a plurality of additional middle-tier servers remote from the client on the path of the CORBA method invocation between the middle-tier server and the database-tier server;

the additional middle-tier servers each creating a transaction policy by translating deployment descriptor files during deployment of the servers;

an interceptor residing on each of the additional middle-tier servers intercepting the IIOP message;

the interceptor residing on each of the additional middle-tier servers extracting the object representing the transaction context from the service context of the IIOP message and reading the method name from the IIOP message;

the interceptor residing on each of the additional middle-tier servers checking the transaction policy for the additional middle-tier server with respect to the server;

the interceptor residing on each of the additional middle-tier servers returning the IIOP message to its original path without completing a control object interpositioning process.

16.   (Original)   The method of claim 13, wherein the interceptor residing on the middle-tier server checks the transaction policy for the middle-tier server with respect to the server before extracting the object representing the transaction context from the service context of the IIOP message and reading the method name from the IIOP message.

10

18349.03/4000.01900

Patent

17.   (Original)   The method of claim 13, wherein the interceptor residing on the middle-tier server checks the transaction policy for the middle-tier server with respect to the server after extracting the object representing the transaction context from the service context of the IIOP message and reading the method name from the IIOP message.

18.   (Original)   The method of claim 12, wherein the transaction policy created on the middle-tier server is created during deployment of the middle-tier server; and,

wherein the transaction policy created on the database-tier server is created during deployment of the database-tier server.

19.   (Original)   The method of claim 12, wherein the transaction policy created on the middle-tier server is created after receipt of the IIOP message to facilitate run-time checking of the deployment descriptor file to determine the transaction policy for the server; and,

wherein the transaction policy created on the database-tier server is created after receipt of the IIOP message to facilitate run-time checking of the deployment descriptor file to determine the transaction policy for the server and to facilitate run-time comparison of the method name with the deployment descriptor file.

11

18349.03/4000.01900